

 码龄8年 暂无认证

16 原创

13万+ 周排名

198万+ 总排名

5万+ 访问

等级

618 积分

29 粉丝

61 获赞

21 评论

238 收藏

私信 关注

参与话题写文章得原力分, 点亮勋章

去发布

搜博主文章

热门文章

利用C语言实现大数加减法 21571

7-1 厘米换算英尺英寸 (基础编程题) 9855

利用阶乘讲普通递归和尾递归 3722

C语言报错警告合集 (转) 3563

快速判断一个数是否是2的幂次 3368

分类专栏

 C语言二三事 9篇

 PAT 2篇

 PTA 5篇

最新评论

位操作符详解与举例 (将指定的位置设为... sixueyi0902: 第一个有问题吧! 首先7的二进制编码为0111, 而3 (bit\_number) 的 ...

快速判断一个数是否是2的幂次 hhhhhhhhhhhhhhhhhhhhhhh: 有一个例外0, 0并不是2的幂次, 但是0&(0-1)==0

习题11-8 单链表结点删除 (20分) 杯中窥人168: 代码是错的....机测wa了

利用函数指针 综合求数组的升序和降序宁cc: swap里面有点问题

利用C语言实现大数加减法 我得269: 怎么用个程序实现加和减

您愿意向朋友推荐“博客详情页”吗?

 强烈不推荐

 不推荐

 一般般

 推荐

 强烈推荐

最新文章

B1018 锤子剪刀布 (20分)

习题11-8 单链表结点删除 (20分)

习题11-7 奇数值结点链表 (20分)

2020年 7篇

2019年 1篇

2018年 9篇

原创

不知道换什么昵称好 于 2018-10-25 02:41:56 发布 3723 收藏 6

分类专栏: C语言二三事 文章标签: 递归 尾递归 阶乘

 C语言二三事 专栏收录该内容

2 订阅 9 篇文章 订阅专栏

阶乘是一个很基本的数学问题,  $n!=n*(n-1)*(n-2)*...*2*1$ , 利用程序来解决阶乘问题比较简单, 可以利用循环或者递归, 这里讲两种 **递归方法**。

第一种: 普通递归:

```
1 int factorial(int n)
2 {
3     if(n==1 || n==0)
4         return 1;
5     else
6         return n*factorial(n-1);
7 }
```

取 $n=3$ 为例, 可以利用CB中的call stack 来观察递归完成回归时是一步一步依次从函数fab(1), fab(2), fab(3)完成回归的。

Nr	Address	Function	File	Line
0		factorial (n=1)	D:\CB TEST\TEST2\main.cpp	15
1	0x4013bc	factorial(n=2)	D:\CB TEST\TEST2\main.cpp	18
2	0x4013bc	factorial(n=3)	D:\CB TEST\TEST2\main.cpp	18
3	0x40137e	main()	D:\CB TEST\TEST2\main.cpp	8

第二种: 尾递归:

```
1 int factorial(int n, int a)
2 {
3     if(n==1 || n==0)
4         return a;
5     else
6         return factorial(n-1, n*a);
7 }
```


同样取 $n=3$ 为例, 可以利用CB中的call stack 来观察尾递归完成回归时是一步从函数fab(1, 6)中跳出完成回归。

Nr	Address	Function	File	Line
0		factorial (n=1, a=6)	D:\CB TEST\TEST2\main.cpp	24
1	0x4013cd	factorial(n=2, a=3)	D:\CB TEST\TEST2\main.cpp	26
2	0x4013cd	factorial(n=3, a=1)	D:\CB TEST\TEST2\main.cpp	26
3	0x401386	main()	D:\CB TEST\TEST2\main.cpp	8

由以上两种递归解决阶乘的例子可以看出在递归回归的时候尾递归和普通递归是不一样的, 普通递归需要沿着一步一步进来的函数一步一步地出去, 而尾递归则可以直接回到主函数。以上也可以说明尾递归比普通递归更能节约函数的堆栈空间。尾递归也更容易转换为与其对等的迭代控制结构。

阅读终点, 创作起航, 您可以撰写心得或摘录文章要点写篇博文。去创作 >

一键收藏

 不知道换什么昵称好 关注

3 6 1

专栏目录

Python递归及尾递归优化操作实例分析 12-20

本文实例讲述了Python递归及尾递归优化操作。分享给大家供大家参考，具体如下： 1、递归介绍 递归简而言之就是自己...

阶乘的尾递归形式 SE\_and\_CS 3558

以尾递归的形式定义计算！

浅谈普通递归和尾递归 weixin\_42048463的博客 2383

[举例(计算阶乘)] #include<stdio.h> /\* 用尾递归思想计算阶乘 \*/ unsigned long Fact(unsigned int n, unsigned long a) { if(n ==...

使用尾递归计算阶乘 weixin\_33918114的博客 2431

通常在计算一个数的阶乘的时候，我们会使用递归这种算法。递归表现出来的书程序调用自身，且具有边界条件、前进段、...

尾递归优化, 以阶乘函数为例 Archimelan的博客 1142

function factorial(n){ if (n &lt;= 1){ return 1;}else{ return n \* factorial(n-1) }} 上面是阶乘函数的常规递归实现 尾递归就是每...

尾递归求阶乘 weixin\_46447561的博客 2918

首先我们很容易想到普通的递归 #include<stdio.h> double f(int n) { double result; if(n==0||n==1) result=1; else result=n\*f(n-...

关于尾递归的使用详解 12-19

这几天看到几篇关于尾递归的文章，之前对尾递归没有多大...从代码层面看，尾递归其实一句话就可以说清楚了： 函数的最...

尾递归.cpp 09-22

/\*int f(int n,int a1,int a2) { if(n) return a1; else return f(n - 1, a2, a1 + a2); }\*/

Python中使用装饰器来优化尾递归的示例 12-24

阶乘函数factorial, 通过把计算值传递的方法完成了尾递归。但是python不支出编译器优化尾递归所以当递归多次的话还是会...

C#中尾递归的使用、优化及编译器优化 12-26

递归运用 一个函数直接或间接的调用自身，这个函数即可叫做递归函数。 递归主要功能是把问题转换成较小规模的子问题...

C 语言尾递归求阶乘.9.1 函数: 尾递归计算阶乘(factorial) weixin\_39614675的博客 703

最简单的递归形式就是把递归调用置于函数的末尾，即正好在return语句之前，这种形式的递归被称为尾递归 (tail recursio...

递归阶乘详解 是阿超 3511

递归阶乘 什么是递归? 直接递归: 方法自身调用自己，当满足一定条件时跳出。以编程的角度来看,递归指的是方法定义中...

递归算法求阶乘 热门推荐 俗人浅见 4万+

利用递归算法，构建一个函数计算15以内的阶乘。15以上的阶乘int类型已经不能存储。 #include<stdio.h> int fact(int); ...

[递归]什么是递归&阶乘 最新发布 YJH000\_的博客 242

一次次的传递，一次次归返便是递归

利用递归函数实现阶乘 anni5211314的博客 1万+

通过递归函数等三种方式实现阶乘

递归和尾递归的区别和原理 ZMyths的博客 4万+

递归和尾递归的区别和实现 基本上大多数C的入门教材里都会说简单的递归，例如求阶乘n! 经典...

C语言以尾递归的形式计算阶乘的一个函数实现的代码 weixin\_44006606的博客 386

如下资料是关于C语言以尾递归的形式计算阶乘的一个函数实现的代码。 #include "facttail.h" int facttail(int n, int a) { if (n < ...


阶乘的尾递归 (Tail Recursion) 写法 (C++) zhangpiu的专栏 1646


一般的递归会导致递归调用栈的产生，深层递归下去将导致效率的下降，然而，部分递归问题是尾调用 (Tail Call) 的特殊...


利用尾递归函数求n的阶乘 03-30


在尾递归函数中，最后一步必须是调用自身，并且不能有其他操作。 以下是求n的阶乘的尾递归函数实现: ``python def fa...


“相关推荐”对你有帮助么?

 非常没帮助

 没帮助

 一般

 有帮助

 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照

©1999-2023北京创新乐知网络技术有限公司

发文章得原力分, 点亮勋章



举报